

Санкт-Петербургский государственный университет  
Кафедра математической теории игр и статистических решений

Савельева Маргарита Михайловна

Выпускная квалификационная работа бакалавра

Эволюционная модель конкуренции  
операторов связи

Направление 010400

Прикладная математика и информатика

Научный руководитель,  
кандидат физ.-мат. наук,  
доцент  
Губар Е.А.

Санкт-Петербург

2016

# Содержание

Введение . . . . .	3
Краткий обзор литературы . . . . .	6
Глава 1. Постановка задачи . . . . .	7
1.1. Основные понятия и определения . . . . .	7
1.2. Эволюционные динамики . . . . .	9
Глава 2. Визуализация процесса принятия решений . . . . .	13
2.1. Алгоритм построения графического изображения динамики системы . . . . .	13
2.2. Результаты работы программы . . . . .	15
2.3. Интерфейс программы . . . . .	16
Глава 3. Модель конкуренции операторов связи . . . . .	19
3.1. Формализация задачи конкуренции сотовых операторов . . . . .	19
Заключение . . . . .	25
Список литературы . . . . .	26
Приложение А. . . . .	27
Приложение В. . . . .	31
Приложение С. . . . .	36

## Введение

С развитием средств коммуникаций, в том числе интернета и мобильных сетей, люди стали быстро получать подробную информацию обо всех предложениях на рынке товаров и услуг. Бурный рост в сфере интернет-торговли, в частности онлайн магазинов и курьерских служб, позволяет людям осуществлять покупки в любых точках планеты. Значимость географического положения производства товара со временем уменьшается. Все это приводит к увеличению ассортимента конкурирующих товаров и услуг и, соответственно, росту конкуренции среди продавцов. В связи с этим на рынке складывается ситуация, при которой даже незначительная разница в цене или качестве продукта может сильно повлиять на спрос в пользу конкурента. Поведение покупателей ставит перед продавцом вопрос: как регулировать цены на товары, учитывая конкуренцию, чтобы избежать потери клиентов? Поэтому задача прогнозирования поведения потребительских предпочтений является одной из наиболее актуальных.

Анализ процесса принятий агентами решений о выборе компаний-продавцов должен учитывать совокупное изменение состояния потребительского рынка. Для описания таких процессов могут быть использованы модели и методы эволюционной теории игр.

Это относительно новый раздел в области динамической теории игр. Базовые понятия эволюционной теории игр сформировались на основе исследований процесса естественного отбора в биологических системах [10]. Данная теория изучает поведение больших групп населения (популяций), состоящих из стратегически взаимодействующих агентов, и оценивает их эффективность.

С развитием эволюционного подхода и расширением его для применения к экономическим моделям, идеи распространения "успешных" стратегий поведения через наследование были преобразованы в модель подражания "успешным" стратегиям. Благодаря такому подходу в последние несколько лет эволюционная теория игр развивается особенно быстро. На-

пример, ее инструменты применяют в задачах логистики и управления для исследования мультиагентных систем, то есть в тех областях, где традиционные представления об агентах и их рациональности не могут быть оправданы.

В предположениях эволюционной теории агенты популяции выбирают свои стратегии исходя из состояния окружающей среды. То есть, лучшая стратегия обеспечивает лучшую адаптацию. Одной из основных задач эволюционной теории игр является прогнозирование "выживаемости" стратегии. В исследованиях, касающихся процесса естественного отбора, "выживаемость" определенной стратегии предполагает реальную выживаемость группы особей, использующих (наследующих) определенное поведение, обусловленное их физическими особенностями. Однако, по мере развития теории была получена возможность экономической интерпретации данных моделей. Под популяцией однородных особей может пониматься большая группа агентов с определенными экономическими целями, а под множеством стратегий – набор экономических действий, выбор которых влияет на достижение указанных целей.

Например, при моделировании поведения потребительского рынка можно говорить о стратегии агента как о выборе агентом одной из конкурирующих компаний в качестве поставщика товаров или услуг. В этом случае "выживаемость" стратегии подразумевает достаточное число клиентов, пользующихся услугами определенной компании продолжительное время.

Аналогичный эволюционный подход может быть применен к задаче прогнозирования поведения абонентов сотовых компаний. Выбор такого подхода обусловлен тем, что множество абонентов представляет собой большую группу лиц, в которой расходы каждого индивида зависят от поведения остальных. Долевое распределение абонентов по представленным на рынке сотовым операторам формирует состояние среды.

Моделирование таких процессов имеет ряд особенностей. Во-первых, изменения состояния системы описываются системой нелинейных диффе-

ренциальных уравнений. Получить их решение в общем виде достаточно сложно, поэтому многие эволюционные задачи требуют численного решения. Во-вторых, для получения представления об общих тенденциях поведения клиентов компаний необходимо рассмотреть большое количество различных начальных условий.

С учетом этих особенностей, целью данной работы являлось получение программного продукта для графического моделирования процесса принятия решений агентами с помощью инструментов эволюционной теории игр с возможностью изменения пользователем различных параметров и начальных условий задачи.

## Краткий обзор литературы

В качестве основных источников для изучения эволюционного подхода в данной работе были использованы такие признанные учебники, как Weibull J. "Evolutionary Game Theory" [11] и Hofbauer J., Sigmund K. "Evolutionary Games and Population Dynamics" [5]. В данных учебниках авторы приводят основные результаты и исследуют нелинейную динамику саморегуляции социального и экономического поведения агентов и тесно связанное с ней взаимодействие между видами в экологических сообществах. Расширение этих подходов и их классификация были представлены позже в учебнике Sandholm W. H. "Population Games and Evolutionary Dynamics" [6]. Работа представляет собой обобщение многих известных результатов и методов эволюционной теории игр, а также подробное описание основных эволюционных динамик. Другим учебником, в котором приведены основные методы эволюционной теории игр является Cressman R. "Evolutionary Dynamics and Extensive Form Games" [3]. В работе представлена адаптация игр в развернутой форме к эволюционному подходу. Способ графического представления динамики наилучших ответов для задач с тремя стратегиями, используемый в данной выпускной квалификационной работе, базируется на материалах указанного ученика.

Концепции, описанные в приведенных выше работах, были применены к реальным задачам медицины и представлены в учебном пособии Колесин И. Д., Губар Е. А., Житкова Е. М. "Стратегии управления в медико-социальных системах" [1]. Исследованию эволюционной теории игр также посвящено большое количество статей. При подготовке к выполнению данной работы наибольшее внимание было уделено статьям Sandholm W. H., Hofbauer J. "Stable Games and their Dynamics" [7], Sandholm W. H., Lahkar R. "The projection dynamic and the geometry of population games" [8], Gubar E. "Construction of different types of dynamics in an evolutionary model of trades in the stock market" [4], в которых представлены дополнительные результаты, касающиеся эволюционного моделирования.

# Глава 1. Постановка задачи

Для создания визуального представления поведения абонентов сетевых компаний была описана математическая модель процесса принятия решений в рамках эволюционной теории игр.

Рассмотрим множество клиентов компаний как популяцию, отвечающую нескольким предположениям:

- число агентов популяции велико, но конечно;
- каждый отдельный агент не оказывает (или оказывает незначительное влияние) на других агентов;
- выигрыш каждого агента зависит выбора стратегий другими агентами;
- каждый агент использует какую-либо чистую стратегию, число которых конечно. Таким образом вся популяция делится на группы агентов, использующих одинаковые стратегии [6].

Выбор стратегии каждым агентом популяции эквивалентен выбору поставщика товаров или услуг среди конкурирующих компаний, при этом вероятность выбора определенной стратегии в каждый момент времени зависит от состояния среды.

## 1.1. Основные понятия и определения

Введем основные обозначения для описания эволюционной игры. Рассмотрим эволюционную игру  $G$ , в которой число агентов популяции конечно и может быть записано как  $N = n(t) = \sum_{i \in K} n_i(t) > 0$ , где  $n_i(t) \geq 0$  — число агентов, использующих чистую стратегию  $i$  из множества чистых стратегий  $K = \{1, \dots, k\}$ . Обозначим через  $\Delta$  - множество смешанных стратегий. Вектор  $x(t) = \{x_1(t), \dots, x_k(t)\}$  характеризует долевое распределение агентов по  $k$  стратегиям, где

$$x_i(t) = \frac{n_i(t)}{n(t)}.$$

Таким образом, состояние популяции  $x$  описывается совместным поведением агентов. Множество всех возможных состояний популяции составляет пространство  $\Delta$ :

$$\Delta = \{x : \sum_{i \in K} x_i = 1, x_i \geq 0, \forall i \in K\}. \quad (1)$$

То есть вектор состояния популяции  $x(t) \in \Delta$ , и следовательно, его можно формально идентифицировать как смешанную стратегию [1].

Пусть  $A$  — матрица выигрышей первого игрока в симметричной игре, описывающая попарное взаимодействие агентов. Тогда средний выигрыш популяции может быть записан в виде:

$$\bar{F}(x) = xF(x),$$

где  $F(x) = Ax^T$  — вектор выигрышей в чистых стратегиях.

В рамках эволюционной игры предполагается, что все агенты популяции "близоруки". То есть выбор агентом чистой стратегии зависит только от текущего состояния популяции.

Также каждый агент в случайный момент времени, независимо от других агентов, имеет возможность пересмотреть свою стратегию. Смена стратегии происходит с вероятностью, зависящей от состояния системы, и только при условии увеличения выигрыша агентом. Для определения правила пересмотра решений используется следующее понятие [6], [11]:

**Определение.** Протокол пересмотра решений — это такое непрерывное по Липшицу отображение  $\rho : R^n \times X \rightarrow R_+^{n \times n}$ . Скалярную величину  $\rho_{ij}(F(x), x)$  называют условной вероятностью переключения с чистой стратегии  $i \in K$  на чистую стратегию  $j \in K$  при заданном состоянии популяции  $x$  и векторе выигрышей  $F(x(t))$ .

**Определение.** Пусть  $G$  эволюционная игра,  $\rho$  протокол пересмотра решений, тогда имитационная динамика для игры  $G$  и протокола  $\rho$  есть

$$\dot{x}_i = \sum_{j \in K} x_j \rho_{ji}(F(x), x) - x_i \sum_{j \in K} \rho_{ij}(F(x), x). \quad (2)$$



Таким образом, изменение доли агентов, использующих чистую стратегию  $i$ , описывается как разность математических ожиданий доли агентов, меняющих чистые стратегии  $j \in K$  на чистую стратегию  $i \in K$  и доли агентов, меняющих стратегию  $i \in K$  на любую из стратегий  $j \in K$ .

**Определение.** Состояние популяции  $x$  называется стационарным тогда и только тогда, когда

$$(F_i(x) - \bar{F}(x)) x_i = 0 \quad (3)$$

для всех стратегий  $i \in K$ .

**Определение.** Стратегия  $x \in \Delta$  называется эволюционно устойчивой стратегией, если для каждого  $y \neq x$  существует некоторое значение  $\bar{\varepsilon}_y$  такое, что

$$xA(\varepsilon y + (1 - \varepsilon)x)^T > yA(\varepsilon y + (1 - \varepsilon)x)^T \quad (4)$$

выполняется для всех  $\varepsilon \in (0, \bar{\varepsilon}_y)$ .

Множество эволюционно устойчивых стратегий является подмножеством множества равновесных по Нэшу стратегий.

## 1.2. Эволюционные динамики

Рассмотрим несколько примеров имитационных динамик, использованных в данной работе для визуального моделирования процесса изменения состояния системы в зависимости от начальных условий.

*Репликативная динамика (Replicator dynamic).* В случае репликативной динамики протокол пересмотра решений определяется как

$$\rho_{ji}(F(x), x) = x_j[F_j - F_i]_+. \quad (5)$$

Функцию  $[F_i(x) - F_j(x)]_+$  в общем виде можно записать как  $\phi(F_i(x) - F_j(x))$ ,

где  $\phi(x) : R \rightarrow R_+$  обладает свойствами:

$$\phi(x) = \begin{cases} 0, & x \in (-\infty, 0), \\ \text{возрастающая функция,} & x \in [0, \infty). \end{cases}$$

Подставляя выражение (5) для протокола пересмотра решений в выражение (2), получаем систему уравнений, описывающих репликативную динамику:

$$\begin{aligned} \dot{x}_i &= \sum_{j \in K} x_j \rho_{ji}(F(x), x) - x_i \sum_{j \in K} \rho_{ij}(F(x), x) = \\ &= \sum_{j \in K} x_j x_i [F_i(x) - F_j(x)]_+ - x_i \sum_{j \in K} x_j [F_j(x) - F_i(x)]_+, \quad i \in K. \end{aligned} \quad (6)$$

Из определения симплекса (1) следует, что  $x_i \geq 0$  для всех  $i \in K$ . Тогда выражение (6) можно преобразовать следующим образом:

$$\begin{aligned} \dot{x}_i &= x_i [F_i(x) \sum_{j \in K} x_j - \sum_{j \in K} x_j F_j(x)]_+ - x_i [\sum_{j \in K} x_j F_j(x) - F_i(x) \sum_{j \in K} x_j]_+ = \\ &= x_i [F_i(x) - \bar{F}(x)]_+ - x_i [\bar{F}(x) - F_i(x)]_+ \end{aligned} \quad (7)$$

Используя свойства функции  $\phi$ , описанные ранее, из системы (7) получаем уравнение репликативной динамики:

$$\dot{x}_i = x_i \hat{F}_i(x), \quad (8)$$

где  $\hat{F}_i(x) = F_i(x) - \bar{F}$  — избыток выигрыша игрока, использующего чистую стратегию  $i \in K$ . В данном случае агенты принимают решение о смене стратегии на основе сравнения своего текущего выигрыша со средним выигрышем популяции.

Аналогичным способом можно вывести все эволюционные динамики.

*Динамика Смита (Smith dynamic).*

$$\dot{x}_i = \sum_{j \in K} x_j [F_i(x) - F_j(x)]_+ - x_i \sum_{j \in K} [F_j(x) - F_i(x)]_+. \quad (9)$$

Распределение агентов по стратегиям осуществляется с помощью попарного сравнения выигрыша агентов с различными стратегиями.

*Динамика Браун-фон Нейман-Нэша*  
(*Brown-von Neumann-Nash dynamic*).

$$\dot{x}_i = [\hat{F}_i(x)]_+ - x_i \sum_{j \in K} [\hat{F}_j(x)]_+. \quad (10)$$

Здесь поведение агентов формируется путем сравнения избытков выигрышей по каждой стратегии.

Правые части системы дифференциальных уравнений (8), (9), (10) удовлетворяют условию Липшица, поэтому согласно теореме о существовании и единственности решения задачи Коши системы этих уравнений для любых заданных начальных условий всегда имеют единственное решение [6].

*Динамика наилучших ответов (Best response dynamic).*

Динамика наилучших ответов предполагает дискретный способ изменения состояния популяции. Каждый агент ведет себя рационально и принимает решение о смене стратегии в пользу наилучшего ответа на текущее состояние системы.

**Определение.** Обозначим через  $BR(x)$  — множество наилучших ответов:

$$BR(x) = \arg \max_{y \in \Delta} yA(x) = \{y \in \Delta : yA(x) \geq zA(x), \forall z \in \Delta\} \subseteq \Delta. \quad (11)$$

Динамику состояния системы можно описать следующим способом:

$$\dot{x} \in BR(x) - x. \quad (12)$$

В данном случае протокол пересмотра решений  $\rho_{ji}(F(x), x) \in BR(x)$  и подчиняется механизму фиктивного разыгрывания [5], [3]. В связи с возможностью существования более чем одного наилучшего ответа, такое правило может быть не единственным. В этом случае агенты популяции принимают решение о смене стратегий в пользу одного из наилучших ответов

с равной вероятностью.

Предполагается, что в некоторые моменты времени агенты некоторого подмножества множества всех агентов популяции имеют возможность изменить стратегию. Так как каждый агент имеет множественный выбор, нельзя вычислить состояние системы  $x$ . Обозначим через  $s_m$  математическое ожидание долевого распределения агентов  $x$ , при заданном количестве моментов выбора  $m$ . Эту величину можно оценить как

$$s_m = \frac{1}{m} \sum_{j=1}^m r_j, \quad (13)$$

где  $r_{m+1} = BR(s_m)$ . При построении графической модели динамики наилучших ответов величина  $s_m$  используется как характеристика состояния системы.

**Замечание.** Обозначим через  $\Delta_3$  – множество смешанных стратегий в эволюционной игре с тремя альтернативными стратегиями.

Тогда в случае игры с тремя стратегиями линии

$$l_i : \{x \in \Delta_3 : (Ax^T)_{i-1} = (Ax^T)_{i+1}\}, \quad (14)$$

где  $(Ax^T)_i$  –  $i$ -ая строка матрицы  $F$ , делят пространство  $\Delta_3$  на три региона. В пределах каждого региона траектория изменения состояния системы сохраняет направление, но меняет его строго на границе [3].

## Глава 2. Визуализация процесса принятия решений

Математическая модель конкуренции операторов сотовой связи характеризуется большим числом параметров. Поэтому в процессе работы большое внимание было уделено способу представления результатов работы программы удобным для пользователя способом. Для этого был разработан и реализован алгоритм графического изображения решения эволюционных задач, описанных упомянутыми ранее уравнениями динамик изменения состояния системы (8), (9), (10), (12).

### 2.1. Алгоритм построения графического изображения динамики системы

Представим алгоритмы визуального представления поведения агентов популяции для четырех имитационных динамик, реализованных в пакете Wolfram Mathematica, в рамках решения эволюционных задач с матрицей попарного взаимодействия агентов размерности  $3 \times 3$ . Перечислим их основные этапы.

*Шаг 1.* Составим систему дифференциальных уравнений (2), где протокол пересмотра решений  $\rho_{ij}(F(x), x)$  описывается уравнением, соответствующим выбранной динамике изменения системы. Находим решение системы (2) для различных начальных условий  $x_1(0)$ ,  $x_2(0)$ ,  $x_3(0)$ , подобранных следующим образом:  $x_1(\cdot)$  принимает значения от 0 до 1 с шагом  $n$ , при этом  $x_2(\cdot)$  принимает значения от 0 до  $1 - x_1$  с шагом  $n$  для каждого  $x_1(\cdot)$ .  $x_3 = 1 - x_1 - x_2$  для всех  $x_1(\cdot)$  и  $x_2(\cdot)$ . В большинстве примерах результатов работы программы значением  $n$  было выбрано 0,1. Установление такого шага в большинстве случаев является оптимальным для графического описания большого числа начальных условий (шестьдесят шесть) без перегрузки изображения.

*Шаг 2.* Далее изобразим проекцию пространства смешанных стратегий  $\Delta = \{x : \sum_{i \in K} x_i = 1, x_i \geq 0, \forall i \in K\}$  в новых координатах как равносторонний треугольник с длиной стороны равной единице и верши-

нами в точках  $a_1$  с координатами  $(0,0)$ ,  $a_2$  с координатами  $(0,1)$  и  $a_3$  с координатами  $(\frac{1}{2}, \frac{\sqrt{3}}{2})$ .

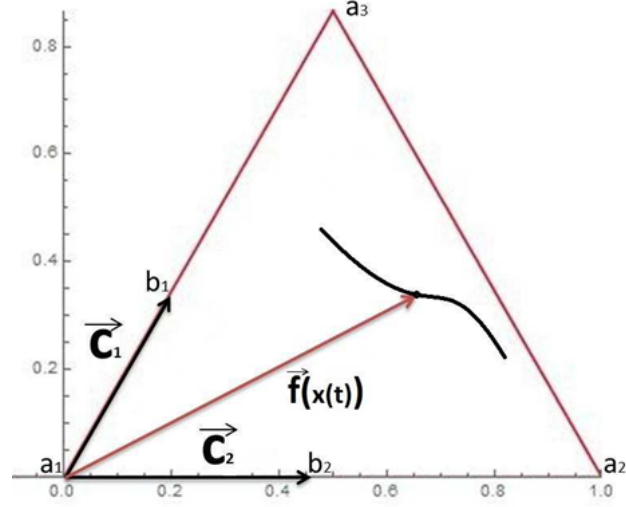


Рис. 1: Графическое изображение проекции пространства  $\Delta_3$

*Шаг 3.* Из определенного ранее симплекса следует, что все изменения состояния системы можно описать двумя функциями  $x_1(t)$  и  $x_2(t)$ . Пусть точки  $b_1$ ,  $b_2$  с координатами зависящими от времени принадлежат отрезкам, составляющим стороны треугольника как показано на рис. 1. Обозначим через  $\vec{c}_1(t)$ ,  $\vec{c}_2(t)$  вектора, удовлетворяющие условиям:  $\vec{c}_1(t) = \overrightarrow{a_1b_1}$  и  $\vec{c}_2(t) = \overrightarrow{a_1b_2}$ . Тогда, если в каждый момент времени  $|\vec{c}_1(t)| = x_3(t)$ ,  $|\vec{c}_2(t)| = x_2(t)$ , то координаты конца вектора, полученного следующим преобразованием:

$$f(x_1, x_2, x_3) = x_1 \begin{pmatrix} 0 \\ 0 \end{pmatrix} + x_2 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + x_3 \begin{pmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{pmatrix}$$

будут описывать состояние популяции в каждый момент времени  $t$ . Из построения следует, что точки  $a_1$ ,  $a_2$  и  $a_3$  соответствуют состояниям системы, когда  $x_1 = 1$ ,  $x_2 = 1$  и  $x_3 = 1$ .

Таким образом, применение преобразования  $f$  к координатам, удовлетворяющим симплексу (1) есть отображение области трехмерного пространства  $x_1 + x_2 + x_3 = 1$  на плоскость.

Для обратного отображения используется преобразование

$$g(y_1, y_2) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + y_1 \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix} + y_2 \begin{pmatrix} \frac{-1}{\sqrt{3}} \\ \frac{-1}{\sqrt{3}} \\ \frac{2}{\sqrt{3}} \end{pmatrix}.$$

*Шаг 4.* Изобразим на графике скорость изменения состояния популяции с помощью встроенной функции DensityPlot. Для этого в каждой точке графика вычислим вектор  $\dot{x}(t)$ . С помощью преобразования, полученного на предыдущем шаге, вычислим  $f(\dot{x}_1(t), \dot{x}_2(t), \dot{x}_3(t))$ . Норма полученного вектора и будет характеризовать скорость изменения системы в каждой точке.

*Шаг 5.* Поиск стационарных состояний системы, то есть решение систем уравнений (3).

## 2.2. Результаты работы программы

Результатом работы каждого раздела программного продукта, отвечающему определенной динамике, является изображение, показывающее изменение состояния системы в рамках любой численно заданной эволюционной задачи.

В качестве примера на рис. 2 приведен результат работы программного комплекса для игры "Камень - ножницы - бумага" [6] в случае динамики Браун - фон Нейман - Нэша. Описание данной игры представлено в приложении С.

На рисунке кривыми черного цвета изображены траектории изменения состояния системы в зависимости от начальных условий, описанных решениями системы уравнений (2).

Области графика разного цвета отображают разную скорость изменения состояния системы и зависят от значения функции  $\|\vec{f}(\dot{x}(t))\|$ . Области красного цвета соответствуют областям наиболее быстрого изменения состояния системы, то есть наиболее высокому значению функции  $\|\vec{f}(\dot{x}(t))\|$ , синего – наиболее медленного.

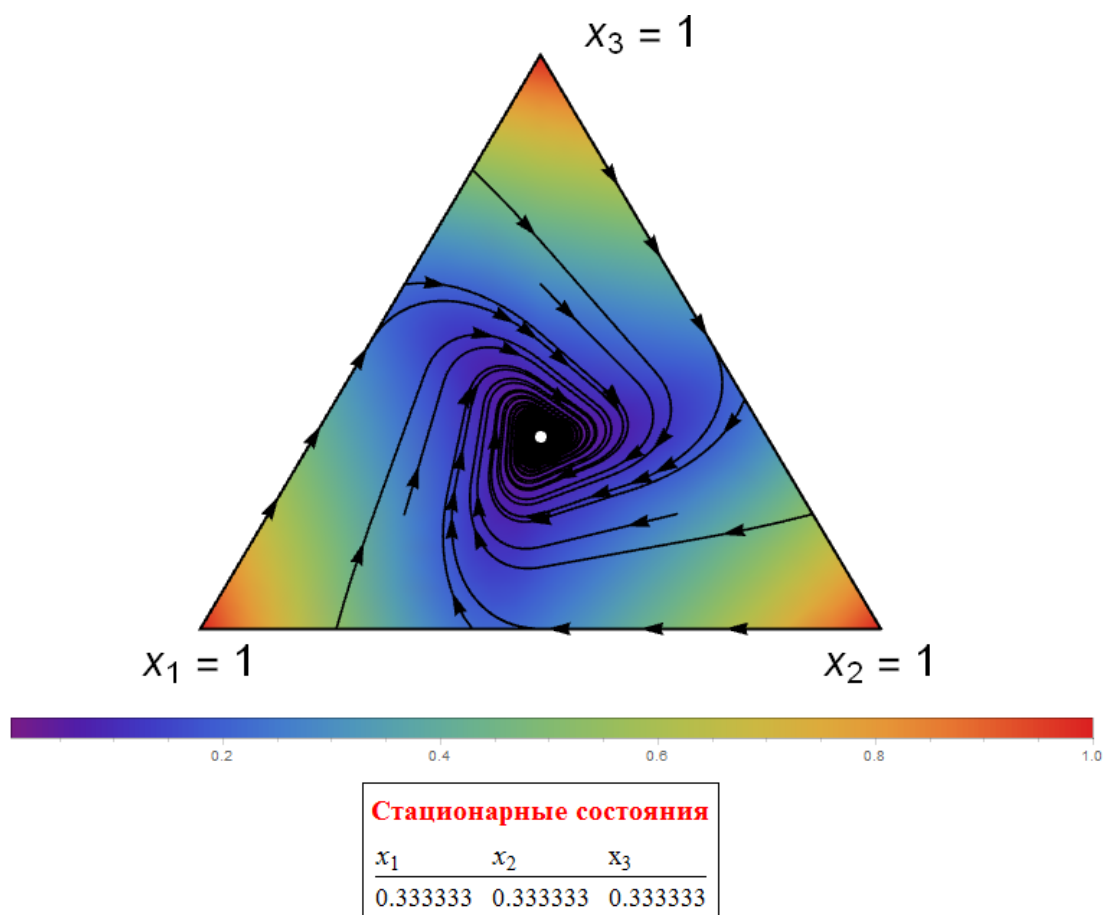


Рис. 2: Процесс изменения состояния системы в игре "Камень - ножницы - бумага", описанный динамикой Браун - фон Нейман - Нэша.

Точками белого цвета отмечены стационарные состояния системы. Координаты таких положений системы генерируются около изображения в виде таблице.

### 2.3. Интерфейс программы

Для удобного анализа зависимости поведения агентов популяции от значений матрицы попарного взаимодействия  $A$  был разработан интерфейс, представленный на рис. 3.



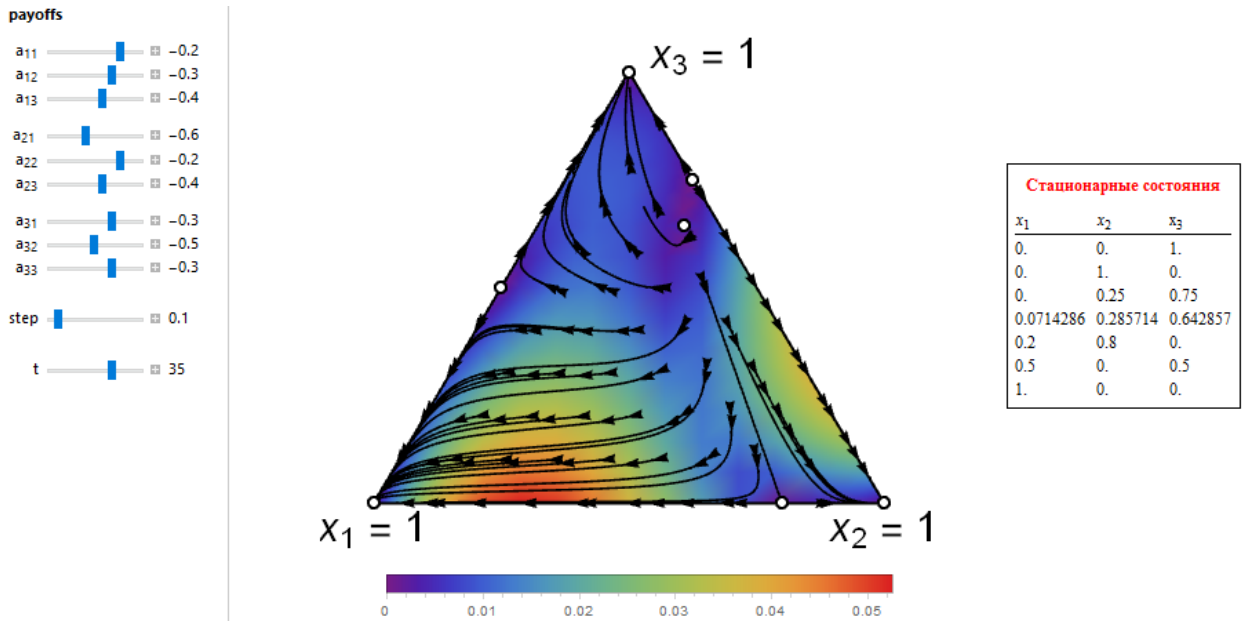


Рис. 3: Интерфейс программы

С помощью помещенных слева от графика ползунков пользователь программного продукта может задавать начальные условия и получать наглядное представление об изменении траектории процесса принятия решений агентами в зависимости от изменения одного или нескольких параметров.

В частности, интерфейс позволяет регулировать значения элементов матрицы  $A$ . В рамках модели конкуренции операторов сотовой связи на экономическом рынке, описанной в следующей главе, предполагается, что значения матрицы  $A$  соответствуют затратам агентов на связь, поэтому интерфейс позволяет изменять значения элементов  $a_{ij}$  в пределах от  $-1$  до  $0$ .

Стоит отметить, что любую матрицу выигрышей можно привести к такому виду, что значения ее элементов будут принадлежать промежутку от  $-1$  до  $0$  [2].

Выбор шага  $n$ , описанного на первом шаге алгоритма, регулируется с помощью ползунка  $step$ . От его значения зависит количество рассматриваемых начальных условий, то есть количество графически изображенных траекторий изменения состояния системы.

Это позволяет пользователю сконцентрировать внимание на траектории изменения долевого распределения агентов по стратегиям с одним

определенным начальным условием или оценить общую тенденцию популяции, меняя шаг  $n$  от 0 до 1.

С помощью ползунка  $t$  можно регулировать максимальное время изменения состояния системы и оценивать возможные положения системы в определенный момент. Изменение этого параметра позволяет пользователю сравнивать скорость достижения состоянием популяции различных стационарных положений в зависимости от начальных условий. Например, на рис. 4 изображены динамики изменения состояния одной и той же системы в разные промежутки времени.

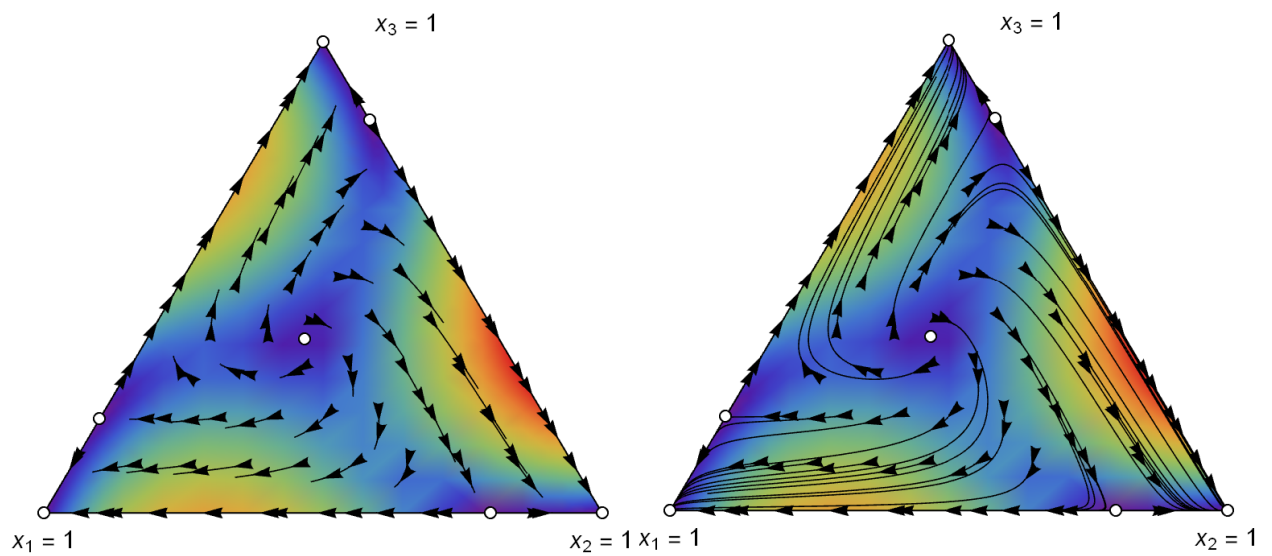


Рис. 4: Разница степени изменения состояния системы при  $t = 5$  и  $t = 35$

В случае использования программы для анализа динамики наилучших ответов пользователь регулирует  $m$  – количество моментов, когда некоторое подмножество множества всех агентов популяции принимает решение о смене стратегии.

## Глава 3. Модель конкуренции операторов связи

Программный продукт, разработанный в ходе работы, может быть использован для визуализации процесса изменения состояния популяции в любой эволюционной игре с тремя заданными стратегиями. То есть подходит для решения задач, описанных любыми матрицами попарного взаимодействия размерности  $3 \times 3$ . Продемонстрируем пример использования программного продукта для решения прикладной задачи графического описания модели конкуренции операторов сотовой связи.

Примеры реализации двух частей программного продукта, описывающих репликативную динамику (6) и динамику наилучших ответов (12) представлены в приложениях А и Б.

### 3.1. Формализация задачи конкуренции сотовых операторов

Рассмотрим динамику поведения агентов популяции, при условии, что на рынке услуг существуют три компании, предоставляющие услуги сотовой связи. В этом случае выбор агентами стратегий эквивалентен выбору сотового оператора.

Клиенты каждой из компаний хотят минимизировать затраты. Предполагаем, что в игре  $G$  матрица  $A$  описывает затраты на услуги сотовой связи взятые со знаком минус. Каждый элемент матрицы  $a_{ij}$  – средние затраты абонента  $i$ –ой компании на общение с абонентами  $j$ –ой компании.

Все абоненты хотят минимизировать свои затраты. В связи с постановкой задачи эта цель эквивалентна максимизации агентами выигрыша в игре  $G$ . Таким образом, затраты всех пользователей мобильной связи зависят от состояния системы.

Каждый клиент компании в некоторый момент времени может принять решение о смене оператора, то есть о смене стратегии, если при этом стоимость услуг связи для него уменьшится. Вероятности таких решения описываются выбором протокола пересмотра решений. В итоге некоторое

подмножество множества всех абонентов меняют свои стратегии в некоторые моменты времени.

Рассмотрим случаи, когда изменение состояния системы описывается одной из четырех динамик (8), (9), (10), (12), приведенных выше. Продемонстрируем работу программного комплекса на конкретном численном примере.

**Задача.** Пусть средняя стоимость звонков на номера того же оператора, что и номер агента составляет 40, 20 и 30 рублей в месяц для 1, 2 и 3 компании соответственно. Звонки на номера любой из конкурирующих компаний составляют 50 рублей в месяц для пользователей номеров 1 и 2 компании и 60 для третьей. Таким образом, матрица попарного взаимодействия  $A$  принимает следующий вид:

$$A = \begin{pmatrix} -0,4 & -0,5 & -0,5 \\ -0,6 & -0,2 & -0,6 \\ -0,5 & -0,5 & -0,3 \end{pmatrix}.$$

Элементы матрицы  $A$  – затраты на мобильную связь, взятые со знаком минус и исчисляемые в сотнях рублей.

**Промежуточные результаты и итоги численного моделирования.** Учитывая условия задачи, составим вектор выигрышей  $F$  в чистых стратегиях:

$$F_1(x(t)) = -0,5 + 0,1x_1(t),$$

$$F_2(x(t)) = -0,6 + 0,4x_2(t),$$

$$F_3(x(t)) = -0,3 - 0,2x_1(t) - 0,2x_2(t).$$

В данном примере функции  $\|\vec{f}(\dot{x}(t))\|$ , характеризующая скорость изменения долевого распределения агентов по стратегиям, принимает значения от 0 до 0,4.

Приведем результаты численного моделирования (см.рис. 5 – 8) для каждой из четырех динамик (8), (9), (10), (12).

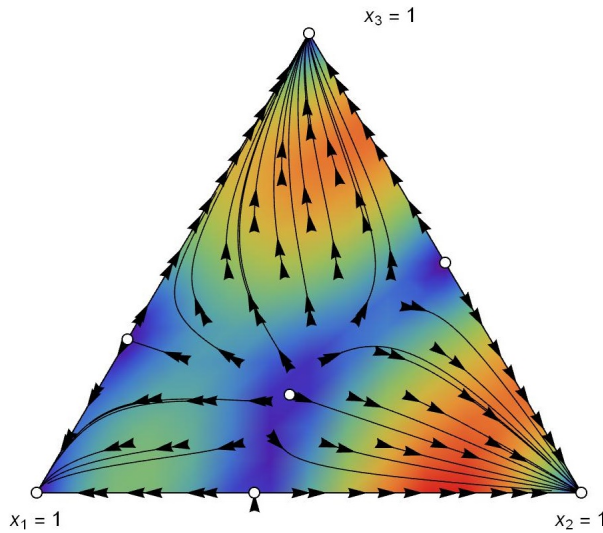


Рис. 5: Репликативная динамика

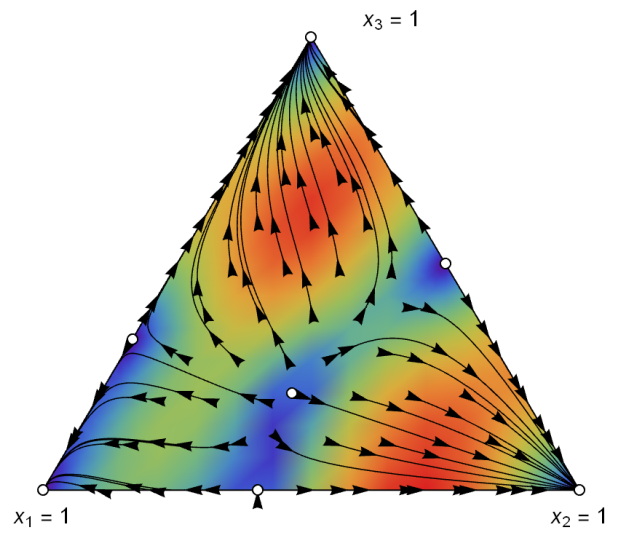


Рис. 6: Динамика Смита

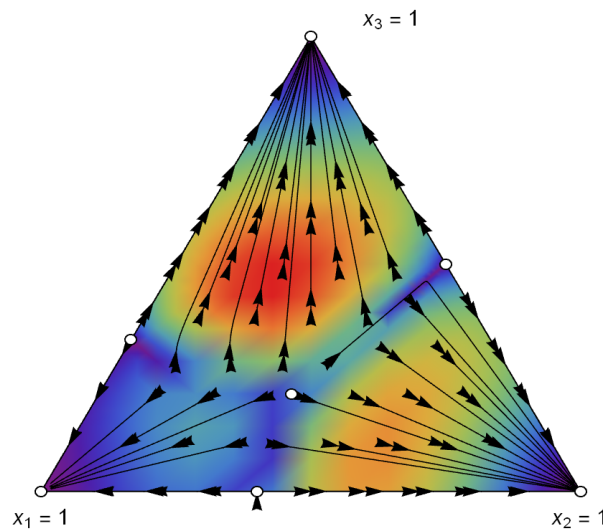


Рис. 7: Динамика Браун - фон Нейман - Нэша

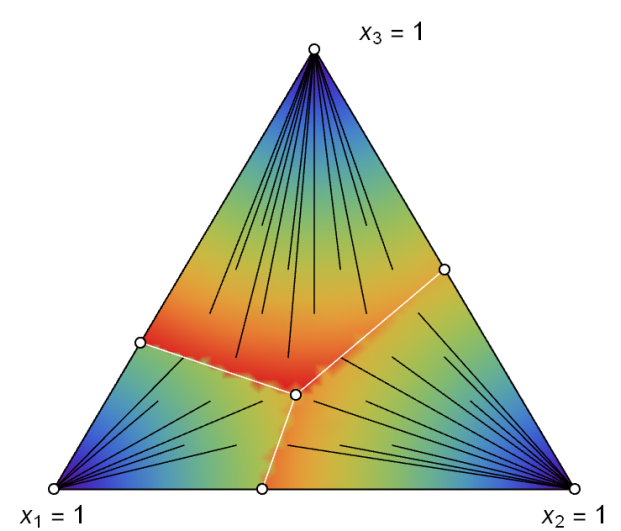


Рис. 8: Динамика наилучших ответов

Полученные рисунки демонстрируют общие тенденции поведения агентов популяции. Как видно из рис. 5 — 8 стационарные точки совпадают. Это связано с тем, что стационарные состояния достигаются тогда, когда все агенты популяции получают одинаковые значения выигрыша [1]. Координаты таких положений не зависят от выбора эволюционной динамики. Таблица координат стационарных положений, полученная в результате работы программы, представлена на рис. 9.

$x_1$	$x_2$	$x_3$
0.	0.	1.
0.	0.5	0.5
0.	1.	0.
0.428571	0.357143	0.214286
0.6	0.4	0.
0.666667	0.	0.333333
1.	0.	0.

Рис. 9: Стационарные положения системы.

Результат работы программы для описания динамики наилучших ответов (рис. 8) имеет дополнительный, отличный от других программ результат. Прямыми белого цвета изображены прямые  $l_i$ , описанные уравнениями (14).

Этим прямым принадлежат границы областей сходимости функций, которые делят пространство  $\Delta_3$  на три региона. Решая уравнения (14), получаем следующее описание прямых  $l_i$ :

$$l_1 = \begin{cases} 0 \leq x_1 \leq 0,75, \\ x_2 = 0,5 - 0,333333x_1, \\ x_3 = 0,5 - 0,666667x_1; \end{cases}$$

$$l_2 = \begin{cases} 0 \leq x_1 \leq 0,666667, \\ x_2 = 1 - 1,5x_1, \\ x_3 = 0,5x_1; \end{cases}$$

$$l_3 = \begin{cases} 0 \leq x_1 \leq 0,6, \\ x_2 = 0,25 + 0,25x_1, \\ x_3 = 0,75 - 1,25x_1. \end{cases}$$

**Выводы.** Оценивая динамику системы с помощью полученных изображений (рис. 5 — 8), можно сделать вывод о зависимости долевого распределения абонентов компаний от ситуации на рынке услуг сотовой связи в начальный момент времени. С течением времени распределение клиентов устанавливается в соответствии с одним из семи стационарных положений, приведенных на рис. 9.

Например, рассмотрим случай, когда в начальный момент времени все агенты популяции являются абонентами одной и той же компании. Эти положения потребительского рынка соответствуют состояниям системы  $(1; 0; 0)$ ,  $(0; 1; 0)$  и  $(0; 0; 1)$ . Из рис. 5 — 8 видно, что эти точки являются устойчивыми положениями динамических систем, описывающих изменения состояния системы. Это значит, что при заданной стоимости услуг, клиентам компаний невыгодно менять своего оператора. Поэтому для привлечения абонентов компаниям, на долю которых в начальный момент приходилась нулевая часть всех абонентов, необходимо снизить цены на услуги.

На рис. 5 — 8 также представлены три области сходимости функций, которые представляют собой решения систем уравнений (8), (9), (10), (12). Каждая область соответствует множеству начальных состояний рынка, при которых абонентам выгодно пользоваться услугами определенного оператора. Оценивая площади этих областей любым возможным способом, можно сделать вывод о том, нуждается ли компания в снижении цен для удержания уже имеющихся клиентов или привлечения новых.

В данном численном примере, при большинстве начальных положений рынка услуг сотовой связи, абоненты из соображений выгоды предпочтут пользоваться услугами второй или третьей компании. Первой компании удастся привлечь клиентов только при условии, что в начальный момент времени доля абонентов, пользующихся ее услугами, больше, чем доля клиентов второй и третьей компании.

Можно сделать вывод о том, что положения системы, соответствующие векторам состояний  $(1; 0; 0)$ ,  $(0; 1; 0)$  и  $(0; 0; 1)$  являются эволюционно устойчивыми.

Рисунки 5 — 8 также демонстрируют наличие состояний потребительского рынка клиентов, не принадлежащих областям сходимости. Эти состояния соответствуют векторам  $(0; 0,5; 0,5)$ ,  $(0,4285; 0,3571; 0,2142)$  и  $(0,66667; 0,33333)$ . Если в начальный момент рынок услуг сотовой связи находится в одном из таких состояний, то при фиксированных ценах, кли-

ентам компаний невыгодно менять оператора связи и экономическая ситуация останется неизменной. Однако при малейшем изменении ситуации на рынке, то есть при даже незначительных изменениях предложенными операторами цен, произойдет перераспределение клиентов среди компаний. Поэтому эти положения системы не являются эволюционно устойчивыми.



## Заключение

В ходе проделанной работы был создан универсальный механизм визуализации процессов принятия решений для четырех эволюционных динамик.

Поскольку существует возможность задания любых начальных условий, полученный продукт можно использовать как для графического представления решения различных задач, так и для получения общего представления о тестируемых эволюционных моделях.

Похожий результат был изложен в 2013 году в работе Hofbauer J., Franchetti F. "An Introduction to Dynamo: Diagrams for Evolutionary Game Dynamics" [9]. Однако, по причине частично скрытого исходного кода и авторской этики результаты этой работы могут быть использованы только для создания демонстрационных изображений и не могут быть расширены для решения некоторых задач.

Программный код, реализованный в результате данной работы, является хорошей базой для дальнейшей работы с эволюционными динамиками. Их дополнение в дальнейшем позволит получать более подробную информацию об интересующих нас аспектах процесса изменения состояния системы. Алгоритм построения пространства  $\Delta_3$ , приведенный во второй главе, может быть использован для описания процессов, подчиняющихся любым эволюционным динамикам. Таким образом, был разработан программный продукт, решающий задачу визуализации решения эволюционных задач с широким спектром возможных расширений.

## Список литературы

- [1] Колесин И. Д., Губар Е. А., Житкова Е. М. Стратегии управления в медико-социальных системах. СПб.: Изд-во С.-Петерб. университета, 2014. 128 с.
- [2] Петросян Л.А., Зенкевич Н.А, Шевкопляс Е.В. Теория игр: учебник. 2-е изд. изд. СПб.: БХВ-Петербург, 2012. 432 с.
- [3] Cressman R. Evolutionary Dynamics and Extensive Form Games. Cambrige: MIT Press, 2003. 312 p.
- [4] Gubar E. Construction of different types of dynamics in an evolutionary model of trades in the stock market // Contributions to Game Theory and Management. 2010. Vol. 3. P. 162–170.
- [5] Hofbauer J., Sigmund K. Evolutionary Games and Population Dynamics. Cambrige: Cambridge University Press, 1998. 323 p.
- [6] Sandholm W. H. Population Games and Evolutionary Dynamics. Cambrige: MIT Press, 2009. 442 p.
- [7] Sandholm W. H., Hofbauer J. Stable Games and their Dynamics // Economic Theory. 2009. P. 1665 - 1693.
- [8] Sandholm W. H., Lahkar R. The projection dynamic and the geometry of population games // Games and Economic Behavior. 2008. Vol. 64. P. 565-590.
- [9] Sandholm W. H., Franchetti F. An Introduction to Dynamo: diagrams for evolutionary game dynamics // Biological Theory . 2013. Vol. 8. P. 167-178.
- [10] Smith J. M. Evolution and the Theory of Games. Cambrige: Cambridge University Press, 1982. 234 p.
- [11] Weibull J. Evolutionary Game Theory. Cambrige: MIT Press, 1995. 365 p.

## Приложение А.

Код программы визуализации процесса изменения состояния системы для репликативной динамики.

```
Simplex2D[{x1_, x2_, x3_}] :=  
x1*{0, 0} + x2*{1, 0} + x3*{1/2, Sqrt[3]/2}  
Simplex2DInverse[{y1_, y2_}] := {1, 0, 0} + y1*{-1, 1, 0} +  
y2*{-1/Sqrt[3], -1/Sqrt[3], 2/Sqrt[3]}  
Payoff[x_, A_] := A.x
```

(\*описание репликативной динамики\*)

```
expectedChange[x_, A_] := (A.x)*x - (x.A.x)*x  
trajectory[initialPoint_List, A_, time_] :=  
Module[{expChange, phaseSolutions},  
expChange = expectedChange[{x1[t], x2[t], x3[t]}, A];  
phaseSolutions = NDSolve[  
{Derivative[1][x1][t] == expChange[[1]],  
Derivative[1][x2][t] == expChange[[2]],  
Derivative[1][x3][t] == expChange[[3]],  
x1[0] == initialPoint[[1]], x2[0] == initialPoint[[2]],  
x3[0] == initialPoint[[3]]}, {x1, x2, x3}, {t, 0, time}];  
Join[x1[t] /. phaseSolutions, x2[t] /. phaseSolutions,  
x3[t] /. phaseSolutions]]
```

```
Manipulate[Module[{initialPoints,  
A = Table[{{a11,a12,a13}, {a21,a22,a23}, {a31,a32,a33}}],  
list, listOfArrows, graphWithTrajectories, background,  
criticalPoints, tMax = tm},
```

(\*начальные условия\*)

```
list = Flatten[Table[Simplex2D[trajectory[N[{initx1,
```

```
initx2, 1 - initx1 - initx2]], A, tMax]], {initx1, 0, 1 +
step/2, step}, {initx2, 0, 1 - initx1, step}], 1];
```

(\*траектория и направление изменения состояния системы\*)

```
listOfArrows = Flatten[Map[Table[{Arrowheads[0.03],
Arrow[{# /. t -> i, # /. t -> (i + 0.1)}]],
{i, 0.2, 3, 2}] &, list], 1];
graphWithTrajectories =
ParametricPlot[Evaluate[list], {t, 0, tMax}, PlotStyle -> Black];
```

(\*скорость изменения состояния системы\*)

```
background = DensityPlot[Norm[Simplex2D[
expectedChange[Simplex2DInverse[{y1, y2}], A]], {y1, 0,
1}, {y2, 0, Sqrt[3]/2}, ImageSize -> {400, 275},
ColorFunction -> "Rainbow", RegionFunction ->
Function[{y1, y2, z}, y2 < Sqrt[3] y1 && y2 < Sqrt[3] (1 - y1)],
BoundaryStyle -> Directive[Black, Thick], Frame -> False,
AspectRatio -> Sqrt[3]/2, PlotLegends -> Automatic,
PlotRange -> {{-0.1, 1 + 0.1}, {-0.1, Sqrt[3]/2 + 0.1}}];
```

(\*стационарные состояния\*)

```
stationaryPoint =
Reduce[{{{Table[{Payoff[{x1, x2, x3}], A},
Payoff[{x1, x2, x3}], A], Payoff[{x1, x2, x3}], A}]} -
Transpose[Table[{Payoff[{x1, x2, x3}], A},
Payoff[{x1, x2, x3}], A], Payoff[{x1, x2, x3}], A}]] == 0) ||
(Payoff[{x1, x2, x3}], A)[[1]] ==
Payoff[{x1, x2, x3}], A)[[2]] &&
x3 == 0) || (Payoff[{x1, x2, x3}], A)[[3]] ==
Payoff[{x1, x2, x3}], A)[[2]] &&
x1 == 0) || (Payoff[{x1, x2, x3}], A)[[1]] ==
```

```
Payoff[{x1, x2, x3}, A][[3]] && x2 == 0) || (x1 == 0 &&
x2 == 0) || (x1 == 1 && x2 == 0) || (x1 == 0 && x2 == 1)) &&
1 == x1 + x2 + x3}, {x1, x2, x3}, Reals,
Backsubstitution -> True];
```

(\*вывод графического изображения\*)

```
Pane[Row[{Show[{background, graphWithTrajectories,
Graphics[listOfArrows], Graphics[{EdgeForm[Thick],
White, (Disk[Simplex2D[#], 0.01] &) /@ ({x1, x2,
x3} /. {ToRules[criticalPoints]})}], Graphics[{
Inset[Style[Row[{Subscript[Style["x", Italic], 3], " = 1"}],
12], {0.65, 0.9}],
Inset[Style[Row[{Subscript[Style["x", Italic], 1], " = 1"}],
12], {0, -0.05}],
Inset[Style[Row[{Subscript[Style["x", Italic], 2], " = 1"}],
12], {1, -0.05}]]}],
PlotRange -> {{-0.1, 1 + 0.1}, {-0.1, Sqrt[3]/2 + 0.1}},
Frame -> False, Axes -> False], Column[{
Text@Style["Стационарные состояния", Red, Bold, 12],
Text[Row[{TableForm[{x1, x2, x3} //.
List[ToRules[N[stationaryPoint]]],
TableHeadings -> {None, {Style[Subscript[
Style["x", Italic], 1], 12],
Style[Subscript[Style["x", Italic], 2], 12],
Style["x3", 12]}}}, TableSpacing -> {1, 1},
TableAlignments -> {Left, Bottom}}], " "],
BaseStyle -> {12}]], Alignment -> Center, Spacings -> 1,
Frame -> True]]], ImageSize -> {1800, 2000}]],
```

(\*организация интерфейса\*)

```
Style["payoffs", Bold], "",
```

```

{{a11, -0.2, "a11"}, -1., 0., 0.1, Appearance -> "Labeled",
ImageMargins -> -2, ImageSize -> Tiny},
{{a12, -0.3, "12"}, -1., 0., 0.1, Appearance -> "Labeled",
ImageMargins -> -2, ImageSize -> Tiny},
{{a13, -0.4, "a13"}, -1., 0., 0.1, Appearance -> "Labeled",
ImageMargins -> -2, ImageSize -> Tiny}, "",
{{a21, -0.6, "a21"}, -1., 0., 0.1, Appearance -> "Labeled",
ImageMargins -> -2, ImageSize -> Tiny},
{{a22, -0.2, "a22"}, -1., 0., 0.1, Appearance -> "Labeled",
ImageMargins -> -2, ImageSize -> Tiny},
{{a23, -0.4, "a23"}, -1., 0., 0.1, Appearance -> "Labeled",
ImageMargins -> -2, ImageSize -> Tiny}, "",
{{a31, -0.3, "a31"}, -1., 0., 0.1, Appearance -> "Labeled",
ImageMargins -> -2, ImageSize -> Tiny},
{{a32, -0.5, "a32"}, -1., 0., 0.1, Appearance -> "Labeled",
ImageMargins -> -2, ImageSize -> Tiny},
{{a33, -0.3, "a33"}, -1., 0., 0.1, Appearance -> "Labeled",
ImageMargins -> -2, ImageSize -> Tiny}, "", "",
{{step, 0.1, "step"}, 0., 1.1, 0.1, Appearance -> "Labeled",
ImageMargins -> -2, ImageSize -> Tiny}, "", "",
{{tm, 3, "t"}, 0, 50, 5, Appearance -> "Labeled",
ImageMargins -> -2, ImageSize -> Tiny},
SaveDefinitions -> True,
SynchronousUpdating -> False,
Alignment -> {Center, Top} ,
ControlPlacement -> Left]

```

## Приложение В.

Код программы визуализации процесса изменения состояния системы для динамики наилучших ответов.

```
Simplex2D[{x1_, x2_, x3_}] :=  
x1*{0, 0} + x2*{1, 0} + x3*{1/2, Sqrt[3]/2}  
Simplex2DInverse[{y1_, y2_}] := {1, 0, 0} + y1*{-1, 1, 0} +  
y2*{-1/Sqrt[3], -1/Sqrt[3], 2/Sqrt[3]}  
Payoff[x_, A_] := A.x;  
  
(*описание динамики наилучших ответов*)  
expectedChange[x_, A_] :=  
Table[FindArgMax[{(Table[{z1, z2, z3}]).A.x, z1 <= 1, z2 >= 0,  
z2 <= 1, z1 >= 0, z3 <= 1, z3 >= 0, z1 == 1 - z2 - z3},  
{z1, z2, z3}] - x];  
BR[r_] := Table[FindArgMax[{(Table[{z1, z2, z3}]).A.r,  
z1 <= 1, z2 >= 0, z2 <= 1, z1 >= 0, z3 <= 1, z3 >= 0,  
z1 == 1 - z2 - z3}, {z1, z2, z3}]];  
  
Manipulate[Module[{list,  
A = Table[{a11, a12, a13}, {a21, a22, a23}, {a31, a32, a33}],  
a, background, line, stationaryPoint, t = tm, n = m},  
  
(*начальные условия*)  
list = Flatten[Table[{initx1, initx2, 1 - initx1 - initx2},  
{initx1, 0., 1 + 0.5, step}, {initx2, 0., 1 - initx1, step}], 1];  
  
(*вспомогательные элементы*)  
a = Table[1, {i, Length[list]}]; h = Table[1, {i, Length[list]}, {j, 2}];  
v = Table[1, {i, Length[list]}, {j, 2}]; m = Table[1, {i, Length[list]}];
```

```

For[k = 0, k <= Length[list] - 1, k++;
p = Table[1, {i, n}, {j, 3}]; d = Table[1, {i, n}, {j, 2}];

(*математическое ожидание положения системы в k-ый момент*)
c = Table[list[[k]]]; p[[1]] = c;
For[i = 1, i <= n - 1, i++;
b := (1/(i - 1)) Sum[BR[p[[j]]], {j, 1, i - 1}]; p[[i]] = b;];

(*преобразование в координаты проекции симплекса*)
For[i = 0, i <= n - 1, i++;
g := Simplex2D[p[[i]]]; d[[i]] = g;
u1 = d[[5]]; h[[k]] = u1;
u2 = d[[6]]; v[[k]] = u2;

(*траектория изменения состояния системы*)
w = ListLinePlot[d, PlotStyle -> Directive[Black],
PlotRange -> {{0, 1}, {-0.01, Sqrt[3]/2 + 0.1}},
AspectRatio -> 1]; a[[k]] = w;]

(*скорость изменения состояния системы*)
background = DensityPlot[
Norm[Simplex2D[expectedChange[Simplex2DInverse[{y1, y2}], A]]],
{y1, 0, 1}, {y2, 0, Sqrt[3]/2}, ImageSize -> {800, 550},
ColorFunction -> "Rainbow", RegionFunction ->
Function[{y1, y2, z}, y2 < Sqrt[3] y1 && y2 < Sqrt[3] (1 - y1)],
BoundaryStyle -> Directive[Black, Thick], Frame -> False,
AspectRatio -> Sqrt[3]/2, PlotLegends -> Automatic,
PlotRange -> {{-0.1, 1 + 0.1}, {-0.1, Sqrt[3]/2 + 0.1}}];
For[k = 0, k <= Length[list] - 1, k++;
q := Graphics[{Arrowheads[0.03], Arrow[{h[[k]], v[[k]]}]}],
Frame -> True, Axes -> False, PlotRange -> {{0, 1},

```



```

{-0.01, Sqrt[3]/2 + 0.1}}, AspectRatio -> 1];
m[[k]] = q]; xx := Table[{x1, x2, x3}];

(*поиск стационарных состояний*)
stationaryPoint =
Reduce[{{{Table[{Payoff[xx, A], Payoff[xx, A], Payoff[xx, A]}] -
Transpose[Table[{Payoff[xx, A], Payoff[xx, A], Payoff[xx, A]}]}]
== 0) || (Payoff[xx, A][[1]] == Payoff[xx, A][[2]] &&
x3 == 0) || (Payoff[xx, A][[3]] == Payoff[xx, A][[2]] &&
x1 == 0) || (Payoff[xx, A][[1]] == Payoff[xx, A][[3]] &&
x2 == 0) || (x1 == 0 && x2 == 0) || (x1 == 1 &&
x2 == 0) || (x1 == 0 && x2 == 1)) &&
1 == x1 + x2 + x3}, {x1, x2, x3}, Reals,
Backsubstitution -> True];

(*построение прямых l*)
dd1 = Reduce[{{{(x1 == 0 && x2 == 0) ||
(Payoff[xx, A][[1]] == Payoff[xx, A][[2]] &&
x3 == 0)) && 1 == x1 + x2 + x3}, {x1, x2, x3},
Reals, Backsubstitution -> True];
dd2 = Reduce[{{{(x1 == 0 && x2 == 1) ||
(Payoff[xx, A][[1]] == Payoff[xx, A][[3]] &&
x2 == 0)) && 1 == x1 + x2 + x3}, {x1, x2, x3}, Reals,
Backsubstitution -> True];
dd3 = Reduce[{{{(x1 == 1 && x2 == 0) ||
(Payoff[xx, A][[3]] == Payoff[xx, A][[2]] &&
x1 == 0)) && 1 == x1 + x2 + x3}, {x1, x2, x3}, Reals,
Backsubstitution -> True];
line := Show[ListLinePlot[(Simplex2D[#] &) /@ ({x1, x2,
x3} /. {ToRules[dd1]}), PlotStyle -> Directive[White]],
ListLinePlot[(Simplex2D[#] &) /@ ({x1, x2,

```

```

x3} /. {ToRules[dd2]}), PlotStyle -> Directive[White]],
ListLinePlot[(Simplex2D[#] &) /@ ({x1, x2,
x3} /. {ToRules[dd3]}), PlotStyle -> Directive[White]]]

```

(\*вывод графического изображения\*)

```

Pane[Row[{Show[background, a, m, line,
Graphics[{EdgeForm[Thick],
White, (Disk[Simplex2D[#], 0.01] &) /@ ({x1, x2,
x3} /. {ToRules[stationaryPoint]}]}], Graphics[{
Inset[Style[Row[{Subscript[Style["x", Italic], 3], " = 1"}],
30], {0.65, 0.9}],
Inset[Style[Row[{Subscript[Style["x", Italic], 1], " = 1"}],
30], {0, -0.05}],
Inset[Style[Row[{Subscript[Style["x", Italic], 2], " = 1"}],
30], {1, -0.05}]}]], Column[{
Text@Style["Стационарные состояния", Red, Bold, 12],
Text[Row[{TableForm[{x1, x2, x3} //.
List[ToRules[N[stationaryPoint]]], TableHeadings -> {None,
{Style[Subscript[Style["x", Italic], 1], 12],
Style[Subscript[Style["x", Italic], 2], 12], Style["x3", 12]}},
TableSpacing -> {1, 1}, TableAlignments -> {Left, Bottom}}], ""],
BaseStyle -> {12}}], Alignment -> Center, Spacings -> 1,
Frame -> True]]], ImageSize -> {1800, 2000}]],

```

(\*организация интерфейса\*)

```

Style["payoffs", Bold], "",
{{a11, -0.2, "a11"}, -1., 0., 0.1, Appearance -> "Labeled",
ImageMargins -> -2, ImageSize -> Tiny},
{{a12, -0.3, "12"}, -1., 0., 0.1, Appearance -> "Labeled",
ImageMargins -> -2, ImageSize -> Tiny},
{{a13, -0.4, "a13"}, -1., 0., 0.1, Appearance -> "Labeled",

```

```

ImageMargins -> -2, ImageSize -> Tiny}, "",
{{a21, -0.6, "a21"}, -1., 0., 0.1, Appearance -> "Labeled",
ImageMargins -> -2, ImageSize -> Tiny},
{{a22, -0.2, "a22"}, -1., 0., 0.1, Appearance -> "Labeled",
ImageMargins -> -2, ImageSize -> Tiny},
{{a23, -0.4, "a23"}, -1., 0., 0.1, Appearance -> "Labeled",
ImageMargins -> -2, ImageSize -> Tiny}, "",
{{a31, -0.3, "a31"}, -1., 0., 0.1, Appearance -> "Labeled",
ImageMargins -> -2, ImageSize -> Tiny},
{{a32, -0.5, "a32"}, -1., 0., 0.1, Appearance -> "Labeled",
ImageMargins -> -2, ImageSize -> Tiny},
{{a33, -0.3, "a33"}, -1., 0., 0.1, Appearance -> "Labeled",
ImageMargins -> -2, ImageSize -> Tiny}"" "",
{{step, 0.1, "step"}, 0., 1.1, 0.1, Appearance -> "Labeled",
ImageMargins -> -2, ImageSize -> Tiny}"" "",
{{m, 3, "m"}, 0, 50, 5, Appearance -> "Labeled",
ImageMargins -> -2, ImageSize -> Tiny},

SaveDefinitions -> True,
SynchronousUpdating -> False,
Alignment -> {Center, Top} ,
ControlPlacement -> Left]

```

## Приложение С.

**Игра "камень - ножницы - бумага".** В игре принимают участие два игрока. Каждый из них независимо от другого выбирает одну из трех стратегий, называемых «камень», «ножницы» и «бумага». Сравнивая выбранные стратегии, игроки определяют победителя. Если стратегии обоих игроков совпадают, выигрыш каждого игрока составляет 0 (ничья), в противном случае побеждает игрок с более сильной стратегией. «Камень» считается сильнее «ножниц», которые, в свою очередь, сильнее «бумаги», которая сильнее «камня». Матрица попарного взаимодействия в данном случае имеет вид:

$$A = \begin{array}{c} \begin{array}{ccc} & \text{К} & \text{Н} & \text{Б} \\ \text{К} & 0 & 1 & -1 \\ \text{Н} & -1 & 0 & 1 \\ \text{Б} & 1 & -1 & 0 \end{array} \end{array}.$$

Если игра ведется внутри популяции, то каждый агент с некоторой вероятностью вступает в игру с другим случайным агентом. Чем больше доля агентов, использующих определенную стратегию, тем выше вероятность встречи с агентом, использующим эту стратегию.